

# Cours d'introduction à l'informatique

## Examen blanc 2024-2025

*Veillez répondre aux questions sur des feuilles séparées sur lesquelles figurent votre nom, prénom et matricule. Les réponses doivent être justifiées. Tous les programmes et fragments de code demandés doivent être rédigés en langage C.*

### Énoncés

- (a) Écrire une fonction prenant en arguments un tableau  $t$  d'entiers, sa taille  $n$ , ainsi qu'un autre entier  $k > 0$ . La fonction doit retourner le nombre d'éléments de  $t$  qui sont multiples de  $k$ . Par exemple, si  $t$  contient  $[5; 10; 4; 8; 4; 3; 7]$  et  $k = 2$ , alors la fonction retourne 4.  
(b) Par la méthode des invariants, démontrer que cette fonction retourne une valeur correcte. Démontrer également que cette fonction se termine.

- En arithmétique, un nombre entier  $n > 1$  est dit *extravagant* si sa décomposition en facteurs premiers contient plus de chiffres (en comptant les puissances) que le nombre de chiffres qui figurent dans l'écriture du nombre lui-même.

Par exemple, le nombre 4 est extravagant, car sa décomposition en facteurs premiers  $2^2$  s'écrit avec deux chiffres alors que 4 n'en nécessite qu'un seul. De même, les nombres  $12 = 2^2 \cdot 3$  et  $110 = 2 \cdot 5 \cdot 11$  sont extravagants, alors que les nombres  $7 = 7$ ,  $125 = 5^3$  et  $10 = 2 \cdot 5$  ne le sont pas.

- (a) Écrire une fonction `nbChiffres` prenant en argument un entier  $n \geq 0$ , retournant le nombre de chiffres qui le composent en décimal. On souhaite que l'implémentation de la fonction `nbChiffres` soit raisonnablement efficace, c'est à dire que sa complexité en temps doit être strictement meilleure que  $O(n)$ .  
(b) Écrire une fonction `extra` prenant en argument un entier  $n > 1$ , retournant 1 si  $n$  est un nombre extravagant et 0 sinon.

Pour résoudre ce problème, on suppose que l'on dispose déjà d'une fonction `unsigned nb_premier(unsigned m)` qui retourne pour tout  $m > 0$  le  $m$ -ième nombre premier. Par exemple, pour  $m = 1, 2$  et  $10$ , cette fonction retourne (respectivement) 2, 3 et 29. La complexité en temps de la fonction `nb_premier` est  $O(1)$ .

Cette fonction doit utiliser la fonction `nbChiffres` implémentée au point (a).

*Note* : Il est permis de programmer une ou plusieurs fonctions supplémentaires de votre choix si votre solution le nécessite.

- (c) Calculer la complexité en temps de la fonction obtenue au point (a).

- (a) Décrire le plus simplement possible l'opération effectuée par la fonction suivante :

```
void f(char *s, unsigned n)
{
    char c;

    if (n <= 1)
        return;

    c = *s;
    *s = *(s + n - 1);
    *(s + n - 1) = c;
    f(s + 1, n - 2);
}
```

- (b) Calculer les complexités en temps et en espace de cette fonction.

- (c) Écrire une fonction réalisant exactement la même opération, mais sans effectuer d'appel récursif.
4. Une calculatrice employant la *Notation Polonaise Inverse (NPI)* fonctionne selon les principes suivants :
- Sa *zone de travail* est constituée d'une pile de nombres réels, initialement vide. La capacité de cette zone est fixée à sa création, et reste constante lors de son utilisation.
  - Lorsqu'on entre un nombre, celui-ci est empilé sur la zone de travail.
  - Lorsqu'on effectue une opération arithmétique, les opérandes de celles-ci sont d'abord dépilés depuis la zone de travail. Ensuite, le résultat de l'opération est empilé.
- Par exemple, si l'on entre successivement les nombres 0,1, 0,2 et 0,3 et si l'on effectue ensuite une opération consistant à multiplier deux nombres, alors la zone de travail contiendra à ce moment [0,1; 0,06]. Si l'on effectue ensuite une opération d'addition, la zone de travail contiendra alors [0,16].
- (a) Définir un type structuré représentant la zone de de travail d'une calculatrice NPI.
- (b) Écrire des fonctions permettant d'effectuer les opérations suivantes de la calculatrice :
- i. Créer une nouvelle zone de travail de capacité donnée.
  - ii. Libérer une zone de travail.
  - iii. Entrer un nombre réel.
  - iv. Additionner deux nombres.
  - v. Multiplier deux nombres.
  - vi. Afficher le contenu (complet) de la zone de travail, sans le modifier.

*Note :* Vous êtes libre de programmer des fonctions ou des types de données supplémentaires si votre solution le nécessite.